Real-time processing gains ground with fast digital multiplier

Refinements in algorithm and hardware cut speed-power product of one-chip device, promising new applications

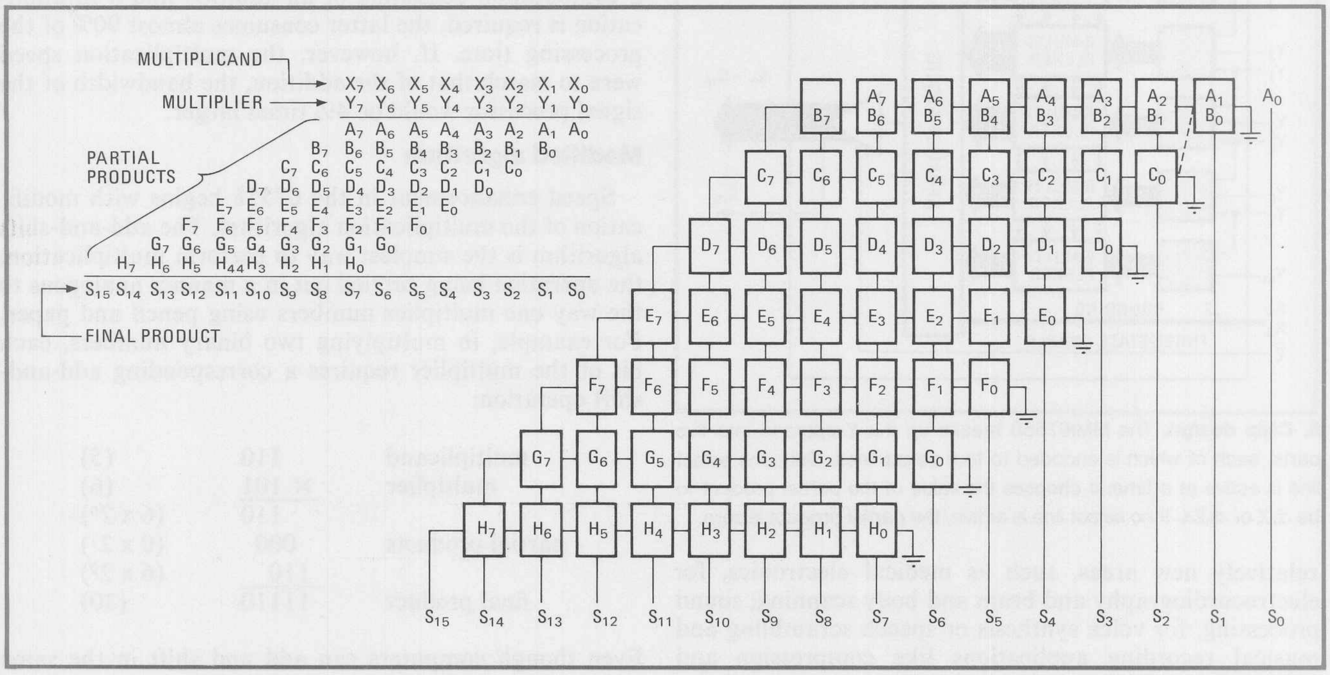
by Shlomo Waser, *Monolithic Memories Inc., Sunnyvale, Calif.,*
and Allen Peterson, *Stanford University, Palo Alto, Calif.*

□ In real-time signal processing, analog methods have held sway despite the advances in digital techniques. But now, fast multiplier chips are helping turn the tide. These chips can speed the complex operations needed for digital treatment, which previously could only be carried out off line using large computers. Functions like autocorrelation and fast Fourier transforms necessary for digital filtering and compression, for example, can now be done in real time by the multipliers and the new, fast analog-to-digital converters. As a result, performance in such fields as telephony, television, and sound reproduction will be greatly enhanced, and digital signal processing will assume the dominance in these areas once held by analog techniques.

The single-chip MMI67558 can multiply two 8-bit numbers in 100 nanoseconds, yet it dissipates only 1 watt. These figures represent a significant breakthrough in speed-power product. The combined attributes of high speed, small size, and low power dissipation destine the 67558 for new roles in signal processing. The classic example of a notch filter 1 hertz wide and 100 decibels deep, which would be pointless to design using tuned circuits, now becomes practical with digital techniques. And spectral analysis, the process of breaking down complex signals into their single-frequency components, is done most accurately using the fast Fourier transform which requires extremely fast multiplication.

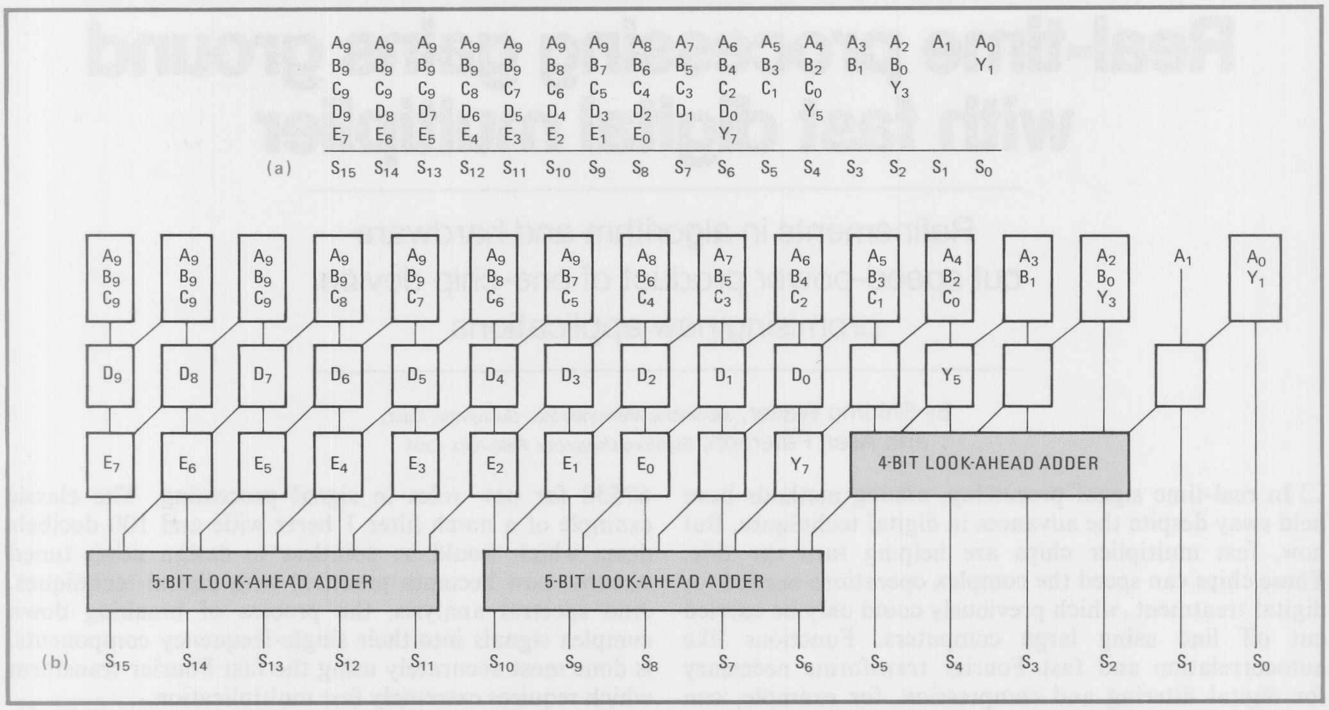
The reduction in cost and size of complex digital components translates into dramatic savings on the system level. The low-power 67558, besides costing far less than the functionally equivalent hard-wired logic, has a smaller power supply requirement, occupies far less circuit board area, and needs no forced-air cooling or heat sinking. Moreover, the reduction in circuit interconnections means an inherently more reliable design.

The 67558 not only has guaranteed entry into the conventional areas of digital signal processing, such as radar and sonar, but it has clear access to a host of


MULTIPLICAND
MULTIPLIER
PARTIAL PRODUCTS
FINAL PRODUCT
X_7 X_6 X_5 X_4 X_3 X_2 X_1 X_0
Y_7 Y_6 Y_5 Y_4 Y_3 Y_2 Y_1 Y_0
A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0
B_7 B_6 B_5 B_4 B_3 B_2 B_1 B_0
C_7 C_6 C_5 C_4 C_3 C_2 C_1 C_0
D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0
E_7 E_6 E_5 E_4 E_3 E_2 E_1 E_0
F_7 F_6 F_5 F_4 F_3 F_2 F_1 F_0
G_7 G_6 G_5 G_4 G_3 G_2 G_1 G_0
H_7 H_6 H_5 H_{44} H_3 H_2 H_1 H_0
S_{15} S_{14} S_{13} S_{12} S_{11} S_{10} S_9 S_8 S_7 S_6 S_5 S_4 S_3 S_2 S_1 S_0

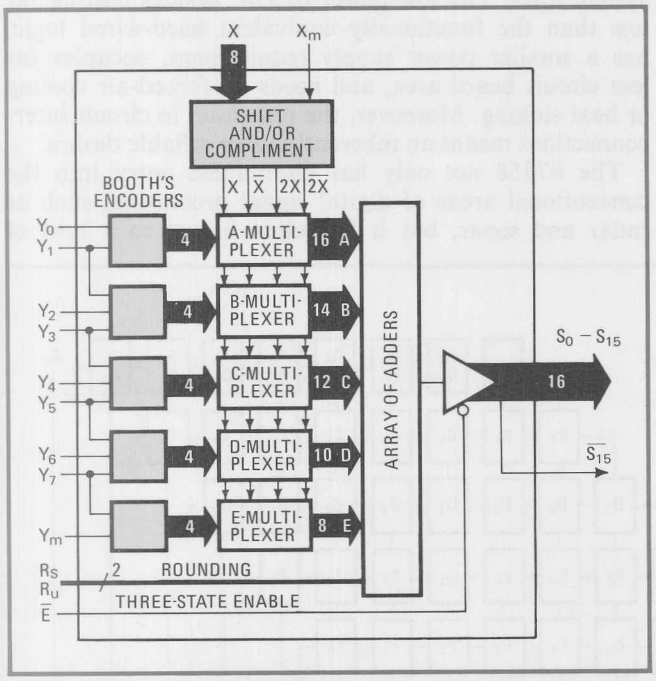
1. Combinatorial multiplier. The double-length product is formed by adding together the eight partial products, each of which is shifted according to the weight of the corresponding multiplier bit. In the scheme at the right, the operation is carried out by 56 binary full adders, and the speed of multiplication is governed by the ripple carry from A_1 through B_7 and on down to S_{15}.

Reprinted from **Electronics**, September 29, 1977 Copyright 1977 by McGraw-Hill Inc. 1221 Avenue of the Americas, New York, N.Y. 10020



(a)
A9 A9 A9 A9 A9 A9 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
B9 B9 B9 B9 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0 Y1
C9 C9 C9 C8 C7 C6 C5 C4 C3 C2 C1 C0 Y3
D9 D8 D7 D6 D5 D4 D3 D2 D1 D0 Y5
E7 E6 E5 E4 E3 E2 E1 E0 Y7
S15 S14 S13 S12 S11 S10 S9 S8 S7 S6 S5 S4 S3 S2 S1 S0
4-BIT LOOK-AHEAD ADDER
5-BIT LOOK-AHEAD ADDER
5-BIT LOOK-AHEAD ADDER
(b)

2. Adder array. Modified Booth's algorithm reduces the number of partial products (a), which are sign-extended to accommodate two's-complement operations. The adder array (b) is arranged in a modified Wallace tree, requiring fewer carry-save adders to sum the partials.



3. Chip design. The MMI67558 breaks up the Y operand into five parts, each of which is encoded to four select lines. Only one select line is active at a time; it chooses the value of the partial product to be ±X or ±2X. If no select line is active, the partial product is zero.

relatively new areas, such as medical electronics, for electrocardiography and brain and body scanning; sound processing, for voice synthesis or speech scrambling and musical recording applications like compression and expansion; and video enhancement, for facsimile transmission, satellite relay of weather photographs, and compression of digital television signals.

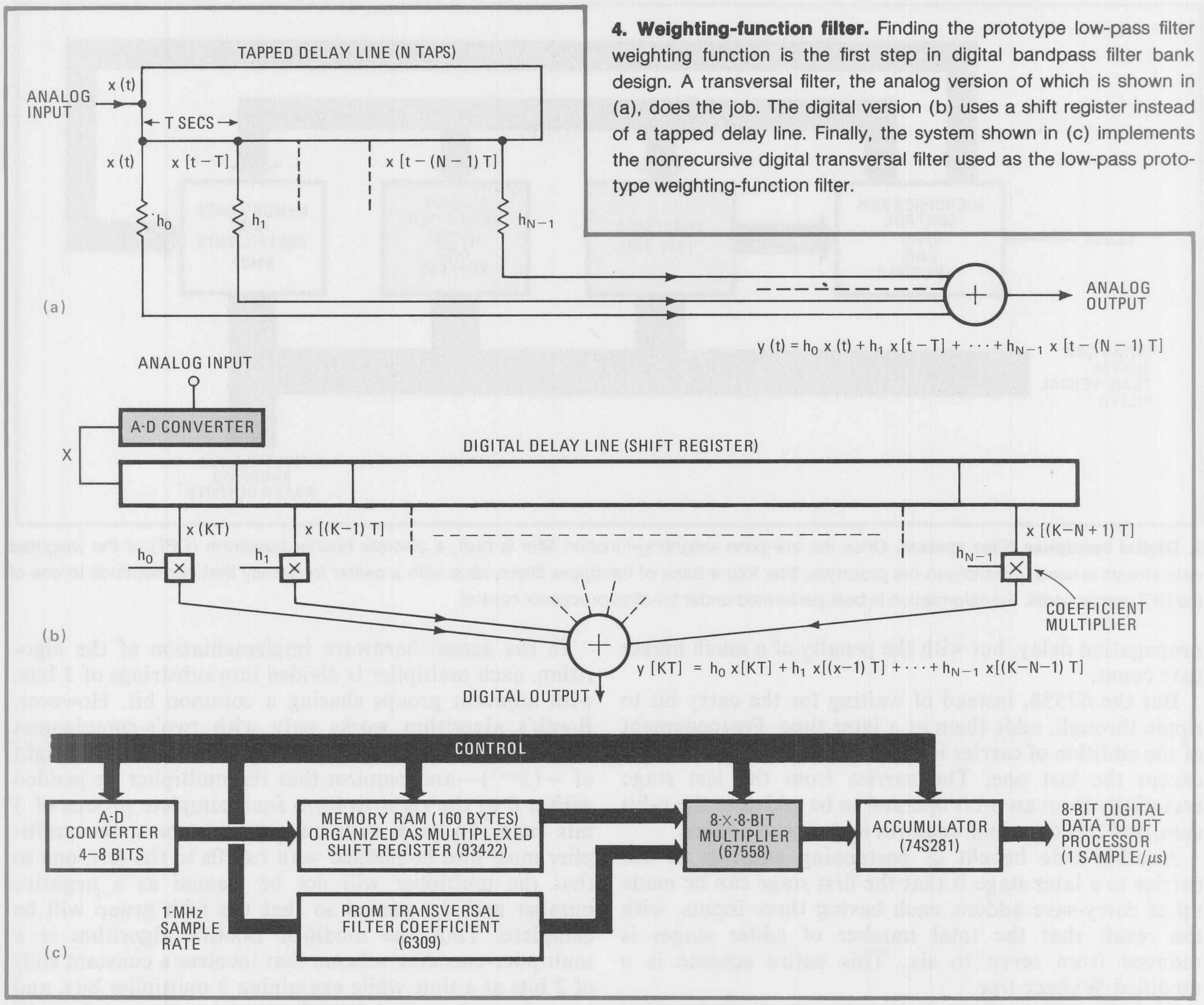
Although it seems like an oversimplification, it is fair to say that the bandwidth in digital signal processing is limited by the speed of multiplication. In the final analysis, all computation reduces to addition and multiplication; and multiplication is traditionally performed by successive addition, such that the calculation of the product of two n-bit numbers requires at least n−1 additions. Thus, if the signal is quantized into 8 bits and a computation consisting of an addition and a multiplication is required, the latter consumes almost 90% of the processing time. If, however, the multiplication speed were to match that of the addition, the bandwidth of the signal processor would be 4½ times larger.

Modified algorithm

Speed enhancement in the 67558 begins with modification of the multiplication algorithm. The add-and-shift algorithm is the simplest way to perform multiplication, the operation being carried out in a manner analogous to the way one multiplies numbers using pencil and paper. For example, in multiplying two binary numbers, each bit of the multiplier requires a corresponding add-and-shift operation:

multiplicand	110	(5)
multiplier	× 101	(6)
	110	(6 x 2^0)
partial products	000	(0 x 2^1)
	110	(6 x 2^2)
final product	11110	(30)

Even though computers can add and shift in the same instruction cycle, n bits of multiplier require at least n such operations. This method, known as sequential multiplication, is performed with adders and shift regis-


TAPPED DELAY LINE (N TAPS)
ANALOG INPUT
x (t)
T SECS
x (t)
x [t − T]
x [t − (N − 1) T]
h_0
h_1
h_{N-1}
(a)
ANALOG OUTPUT
$y(t) = h_0\,x(t) + h_1\,x[t-T] + \cdots + h_{N-1}\,x[t-(N-1)T]$
ANALOG INPUT
A-D CONVERTER
X
DIGITAL DELAY LINE (SHIFT REGISTER)
x (KT)
x [(K−1) T]
x [(K−N + 1) T]
h_0
h_1
h_{N-1}
COEFFICIENT MULTIPLIER
(b)
$y[KT] = h_0\,x[KT] + h_1\,x[(x-1)T] + \cdots + h_{N-1}\,x[(K-N-1)T]$
DIGITAL OUTPUT
CONTROL
A-D CONVERTER 4–8 BITS
MEMORY RAM (160 BYTES) ORGANIZED AS MULTIPLEXED SHIFT REGISTER (93422)
8-X-8-BIT MULTIPLIER (67558)
ACCUMULATOR (74S281)
8-BIT DIGITAL DATA TO DFT PROCESSOR (1 SAMPLE/μs)
1-MHz SAMPLE RATE
PROM TRANSVERSAL FILTER COEFFICIENT (6309)
(c)

4. Weighting-function filter. Finding the prototype low-pass filter weighting function is the first step in digital bandpass filter bank design. A transversal filter, the analog version of which is shown in (a), does the job. The digital version (b) uses a shift register instead of a tapped delay line. Finally, the system shown in (c) implements the nonrecursive digital transversal filter used as the low-pass prototype weighting-function filter.

ters and is adequate for low-speed multiplication.

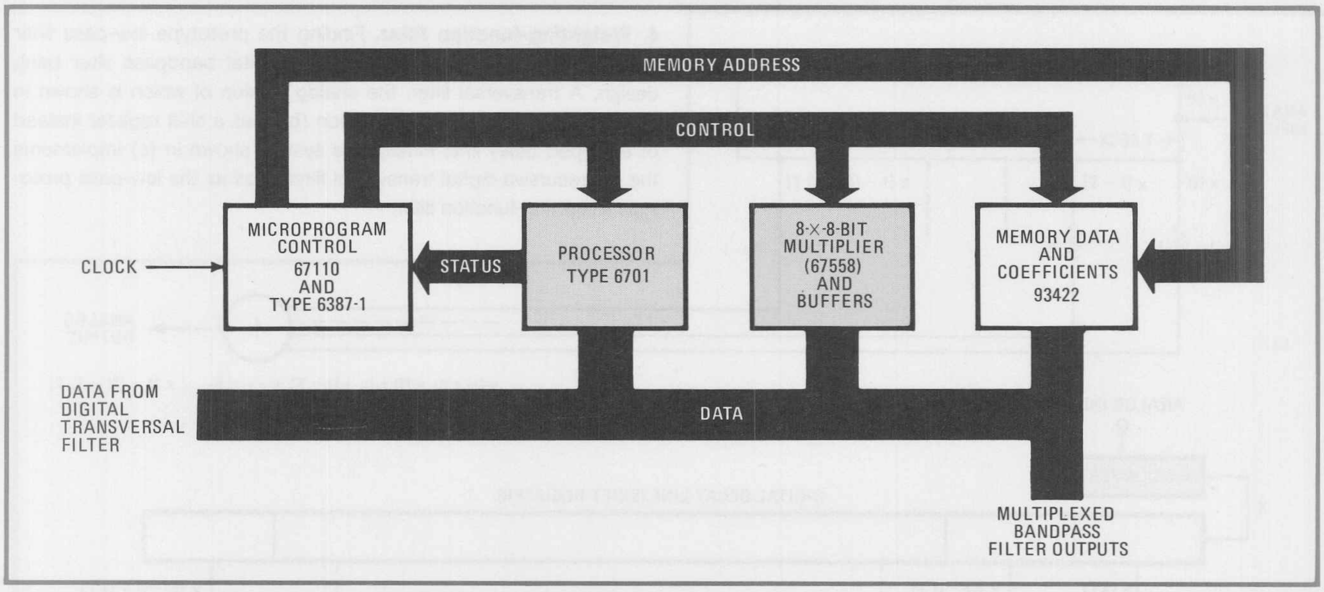
High-speed multiplication calls for a combinatorial approach. In this approach, the partial products are formed simultaneously and then added in one operation. Each successive bit in the partial product is determined by an AND operation of successive multiplicand bits with a single multiplier bit. This procedure is related to the add-and-shift algorithm, since a 0 multiplier produces a zero partial product and a 1 multiplier simply duplicates the multiplicand in the partial product. The actual shifting is performed by the interconnections of the adders used to sum the partials, as shown in Fig. 1.

For the sake of simplicity, the speed of multiplication is best expressed as a function of logic-gate propagation delays, while the power dissipation is directly related to the total number of gates. Referring to Fig. 1, the speed of multiplication is governed by the delays from A_1 to B_7 to S_{15}. This path consists of 14 adder stages, each of which has 4 gate delays; with another gate delay for the generation of partial products, the total is 57. The gate count results from 64 AND gates that generate the partial products and from the 56 binary adders. Since each adder contains 10 gates, total count for the combinatorial multiplier described is $64 + (56 \times 10) = 624$ gates.

INSTRUCTIONS FOR CARRYING OUT MODIFIED BOOTH'S ALGORITHM

Bit			Operation
2^1 Y_{i+1}	2^0 Y_i	2^{-1} Y_{i-1}	
0	0	0	add zero (no string)
0	0	1	add multiplicand (end of string)
0	1	0	add multiplicand (a string)
0	1	1	add twice the multiplicand (end of string)
1	0	0	subtract twice the multiplicand (beginning of string)
1	0	1	subtract the multiplicand (−2X or +X)
1	1	0	subtract the multiplicand (beginning of string)
1	1	1	subtract zero (center of string)

The combinatorial multiplier, though significantly faster than the sequential type, has room for improvement. The carry bits that are passed among the adders slow down the operation, and a technique known as carry-look-ahead, which keeps track of carry bits while not adding them, is often used to reduce the carry-


MEMORY ADDRESS
CONTROL
CLOCK
MICROPROGRAM CONTROL 67110 AND TYPE 6387-1
STATUS
PROCESSOR TYPE 6701
8-×-8-BIT MULTIPLIER (67558) AND BUFFERS
MEMORY DATA AND COEFFICIENTS 93422
DATA FROM DIGITAL TRANSVERSAL FILTER
DATA
MULTIPLEXED BANDPASS FILTER OUTPUTS

5. Digital bandpass filter system. Once the low-pass weighting-function filter is built, a discrete Fourier transform (DFT) of the weighted data stream is used to transform the prototype filter into a bank of bandpass filters, each with a center frequency that corresponds to one of the DFT components. Transformation is best performed under bit-slice processor control.

propagation delay, but with the penalty of a much higher gate count.

But the 67558, instead of waiting for the carry bit to ripple through, adds them at a later time. Postponement of the addition of carries is applied to all the adder stages except the last one. The carries from the last stage essentially form an n-bit operand to be added to the n-bit sum, and this operation requires look-ahead adders.

Another side benefit of postponing addition of the carries to a later stage is that the first stage can be made up of carry-save adders, each having three inputs, with the result that the total number of adder stages is reduced from seven to six. This entire scheme is a modified Wallace tree.

The tree reduces both the number of gate delays and the gate count. Contributors to the propagation delay include 1 gate that generates partial products, 6 carry-save-adder delays of 4 gates each, and 2 stages of carry-look-ahead with 3 gates each, for a total of 31 gate delays. The gate count is made of 64+48 carry-save adders+2 carry-look-aheads. Each of the latter has 30 gates, giving a total count of $64+(48\times 10)+(2\times 30) = 604$ gates.

One other technique is used to increase speed. The approach, the modified Booth's algorithm (first used in the IBM 360), reduces the number of partial products by half; consequently, the number of carry-save-adder stages, and hence the gate count, are also reduced. Basically, Booth's algorithm allows the multiplication operation to skip over any contiguous string of all 1s and all 0s, rather than form a partial product for each bit. Skipping a string of 0s is straightforward, but in skipping over a string of 1s, the following property is put to use: a string of 1s can be evaluated by subtracting the weight of the right-most 1 from the modulus. (The modulus of an n-bit word is 2^n, and the weight of any nth bit is 2^{n-1}, counting from the right.) For example, the value of the binary string 11100 computes to $2^5-2^2=28$.

In the actual hardware implementation of the algorithm, each multiplier is divided into substrings of 3 bits, with adjacent groups sharing a common bit. However, Booth's algorithm works only with two's-complement numbers—the most significant bit of which has a weight of $-(2^{n-1})$—and requires that the multiplier be padded with a 0 to the right to form four complete groups of 3 bits each. To work with unsigned numbers, the multiplier must also be padded with two 0s to the left, one so that the multiplier will not be treated as a negative number and the second so that the fifth group will be complete. Thus the modified Booth's algorithm is a multiplier-encoding scheme that involves a constant shift of 2 bits at a time while examining 3 multiplier bits, and it produces five partial products rather than eight—the fifth partial product being a consequence of the fact that Booth's algorithm only handles two's-complement numbers. All the possible permutations, as shown in the table, form instructions that prevent unnecessary computation for zero partial products.

The hitch in the modified algorithm is that, as shown in the table, a subtractor is required. The subtraction is implemented by adding the two's complement of the number to be subtracted. In taking the two's complement, the number is first complemented and 1 is then added to the least significant bit.

From the table, it can be seen that the Y_{i+1} can simply be added to the LSB of the partial product. If bit $Y_{i+1} = 0$, no subtraction is called for and adding 0 changes nothing. On the other hand, if bit $Y_{i+1}=1$, then the proper two's complement is performed by adding 1 to the LSB. Of course, in the two's complement, the sign bit must be extended to the full width of the final result, as shown by the repetitive adders in Fig. 2a.

The scheme of the modified algorithm is shown in Fig. 2b. The gate delay is made up of 2 gates decoding the 3-bit multiplier instruction, 2 gates for selecting shift or double shift, and 2 carry-save-adder stages and 3 stages of carry-look-ahead, yielding a total of 21 gate delays. The gate count is made up of 5 encoders, 32 multiplexers, 40 carry-save adders, and 3 carry-look-aheads for a total of $(5\times5)+(32\times5)+(40\times10)+(3\times30) = 675$ gates. Thus, for a small penalty in gate count, the speed multiplication is improved, and the output data is also available either signed or unsigned.

Hardware implementation

In implementing the multiplication algorithm, several techniques contribute to the reduced power dissipation in the 67558. First, computer-aided circuit design was used to determine noncritical signal paths—those with lower speed requirements—slowing down the gates in these paths lowers the power dissipation.

Second, the number of devices required for AND-OR-invert gates in the multiplexer section of the circuit was minimized: the logical equation $\overline{C}=\overline{A}\cdot B$ is realized by a single transistor, where C is the collector, B is the base, and A is the emitter. With fewer transistors, each multiplexer dissipates less power.

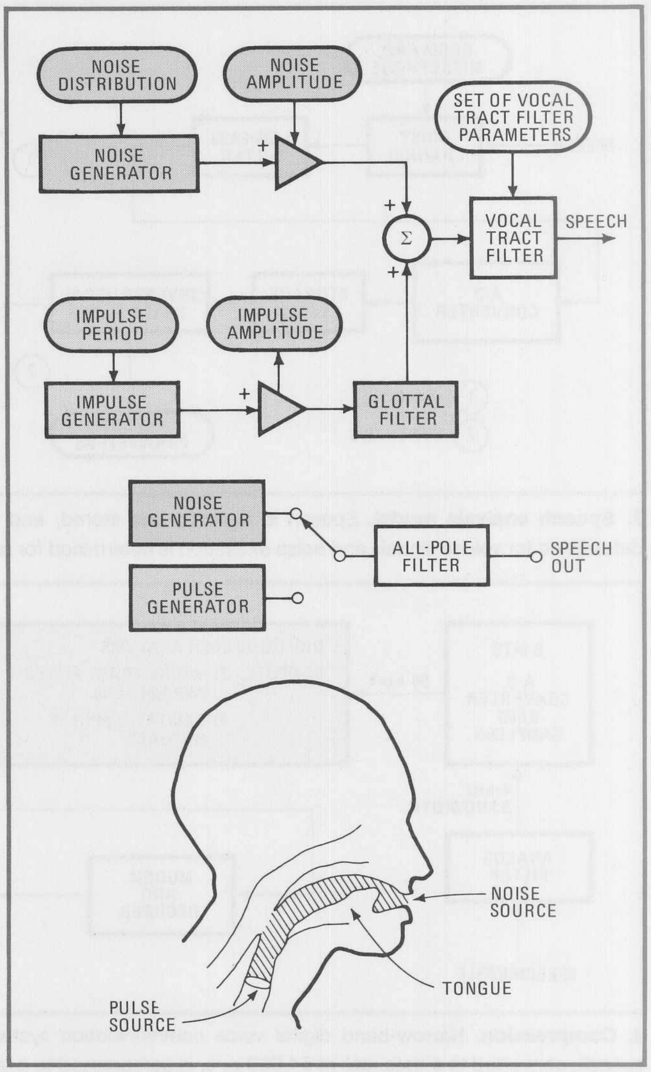
Finally, power is reduced in the implementation of exclusive-OR gates. Although transistor-transistor logic is most suitable for NAND gates, they build an inordinately extensive exclusive-OR circuit. Typically, the problem is circumvented with a circuit that resembles an AND followed by an OR; but this approach produces output voltages that are too high, and level translators must then be used to readjust thresholds. The 67558, however, does not use level translators—instead, the input threshold voltages of subsequent stages are matched to the exclusive-OR outputs. The final design of the 67558 is diagrammed in Fig. 3.

Digital signal processing

In determining the bandwidth limits in the digital processing of analog signals, the first rule is that the sampling rate must be at least twice that of the highest frequency component of the analyzed waveform. In speech processing, for example, since the highest pitch generated by the vocal chords is about 4 kilohertz, a sampling rate of 8 kHz is required. Therefore a sample must be taken at least as often as every 125 microseconds to fulfill the requirement.

A figure for the computational speed of real-time voice processing can be arrived at depending on the operation to be performed. To compute the autocorrelation function for 256 samples of a speech waveform, for example, each sample value is multiplied by 256 other samples. Therefore, in a period of 125 μs, 256 multiplications are required, and thus each multiplication must be peformed in less than 500 ns.

The number of bits of resolution in signal processing is of concern at the time of analog-to-digital conversion and during computations. When an analog signal is converted to a digital word, the signal-to-noise ratio is critical. A poor ratio can result in clipping of the waveform, which, when converted, introduces phantom frequency components. For example, if a waveform at one point has a level of 160 millivolts and each quantizing level is 10 mv, 4-bit resolution is sufficient for the a-d conversion. If the resolution of the system is extended to 8 bits, a maximum noise level or interfering signal of $2^8\times10$ mv = 2.56 volts can be tolerated.

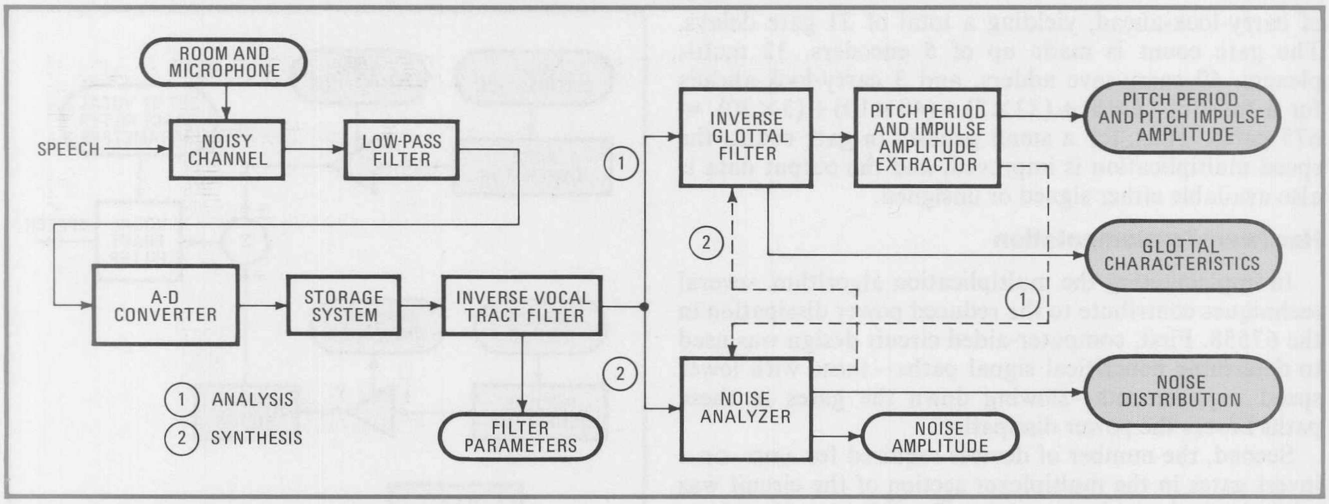


6. Speech mechanism. Understanding the mechanism of speech is key to voice processing. In speech synthesis, the vocal tract becomes a time-varying digital filter excited by pitch impulses (to simulate voiced sounds) or random noise (unvoiced sounds).

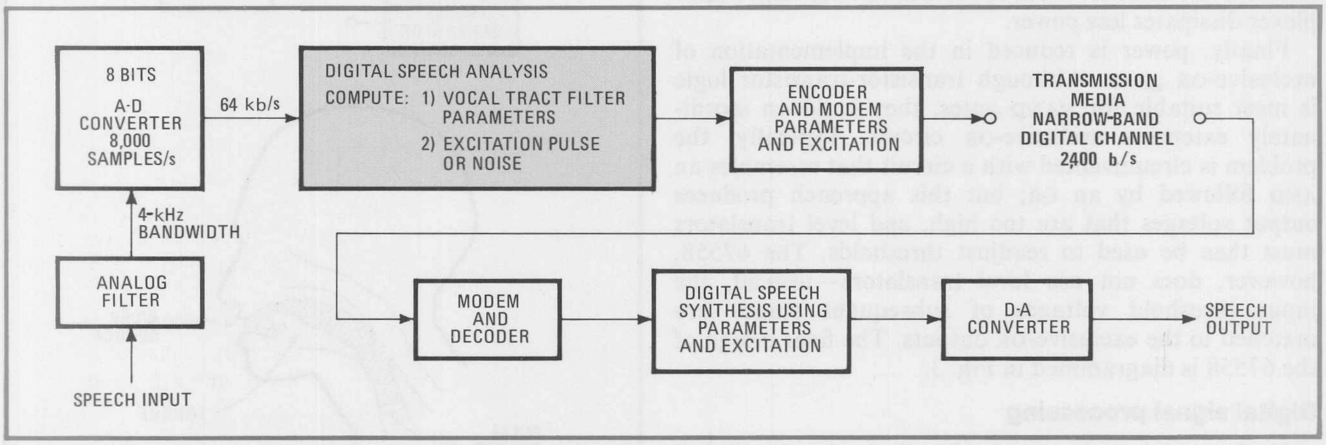
The second place that the number of bits is significant is during the computation. In discrete time analysis, infinite precision of the filter coefficients is usually assumed. In carrying out arithmetic operations, however, the number of bits grows as the calculation proceeds—multiplication may double the word length of the operands, and products must therefore be rounded. Using fixed-point arithmetic with real-time hardware requires careful analysis of operations: too many bits adds to expense and may convey a false sense of accuracy, whereas too few bits results in erroneous analysis.

For full precision in multiplication, each time the product is used as an operand, the number of required bits is doubled. Multiplication of two 8-bit numbers requires 16 bits for full precision, and operating on the result requires a 16-bit multiplier, which in turn generates a 32-bit product.

When using fractional numbers, the full precision is



7. Speech analysis model. Speech is digitized and stored, and vocal-tract filter coefficients are then computed. Next, pitch period is determined for voiced signals and noise excitation is determined for unvoiced signals.


8 BITS
A-D CONVERTER 8,000 SAMPLES/s
64 kb/s
DIGITAL SPEECH ANALYSIS
COMPUTE: 1) VOCAL TRACT FILTER PARAMETERS
2) EXCITATION PULSE OR NOISE
ENCODER AND MODEM PARAMETERS AND EXCITATION
TRANSMISSION MEDIA NARROW-BAND DIGITAL CHANNEL 2400 b/s
4-kHz BANDWIDTH
ANALOG FILTER
SPEECH INPUT
MODEM AND DECODER
DIGITAL SPEECH SYNTHESIS USING PARAMETERS AND EXCITATION
D-A CONVERTER
SPEECH OUTPUT

8. Compression. Narrow-band digital voice communication system shows the advantages of digital over analog processing. Digitized speech, converted to a data rate of 64,000 b/s, is compressed to a rate of 2,400 b/s to reduce the cost of transmission.

not always needed. Instead, a rounding scheme provides the best 8-bit fractional number from the full 16-bit product. Rounding in the binary number system is similar to the scheme used with decimal numbers whereby 0.5 is added to the part to be discarded and the number is then truncated.

Bandpass filter

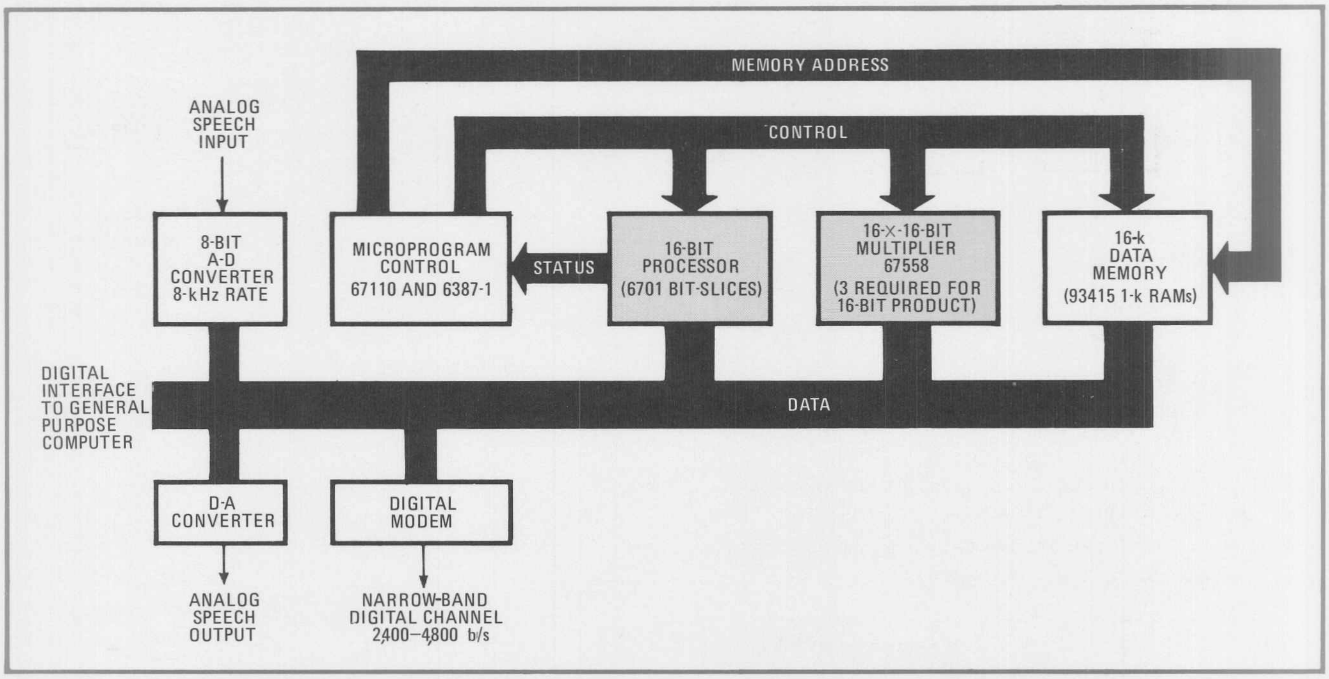
Digital processing techniques can be applied directly to the real-time domain. Bandpass filters, for example, can be assembled efficiently using finite impulse response filters and fast Fourier transform computers. Basically, the design of banks of such filters starts with a low-pass prototype digital filter that derives a weighting function. A discrete Fourier transform of the weighted data stream then transforms the low-pass prototype into a bank of bandpass filters. Each of these filters has a center frequency that corresponds to one of the transform frequency components.

The prototype is a digital version of an analog transversal filter. A standard configuration for the latter consists of a tapped delay line, a set of weighting resistors, and a summer, as shown in Fig. 4a. The output signal is formed by summing the delayed and weighted input signals, according to the equation shown.

To make a digital version, the input signal is first sampled and quantized into a digital form as shown in Fig. 4b. An n-stage digital delay line—or shift register—holds successive numbers (usually in binary format) generated by the a-d converter and shifts the sequence by one state when a clock pulse is applied every T seconds.

The digital filter is nonrecursive, since each output number depends only on the input sequence. Weighted sums of the output sequences are not involved, as they are in the recursive digital filter. Figure 4c gives a system configuration for the transversal filter operations that yields the low-pass prototype.

To complete the design of the multichannel bandpass filter, the n-point discrete Fourier transform of the digital output samples from the transversal filter must be computed. While this computation can be built up with hard-wired medium-scale integrated building blocks, a more flexible system uses a microprocessor. Only minor changes in architecture need then be made to accompany a different number of channels in the bank. Bipolar 4-bit slices such as the 6701 or 2901, together with program controllers like the 67110 or 2911, work well and are


MEMORY ADDRESS
ANALOG SPEECH INPUT
CONTROL
8-BIT A-D CONVERTER 8-kHz RATE
MICROPROGRAM CONTROL 67110 AND 6387-1
STATUS
16-BIT PROCESSOR (6701 BIT-SLICES)
16-X-16-BIT MULTIPLIER 67558 (3 REQUIRED FOR 16-BIT PRODUCT)
16-k DATA MEMORY (93415 1-k RAMs)
DIGITAL INTERFACE TO GENERAL PURPOSE COMPUTER
DATA
D-A CONVERTER
DIGITAL MODEM
ANALOG SPEECH OUTPUT
NARROW-BAND DIGITAL CHANNEL 2,400–4,800 b/s

9. Real-time microprogrammed speech processor. Full-blown speech processing is carried out by this system, which uses bit-slice processors, fast RAM, and 67558 multipliers. Analog output, narrow-band digital output, and interface with a computer are provided.

relatively simple to arrange, as shown in Fig. 5.

The filtering and processing techniques thus developed can be applied to the area of speech analysis and synthesis. Research in this area has led to the introduction of voice encoding and other digital techniques into telephone communications, and these techniques can achieve great savings as the cost of digital hardware declines.

Real-time implementation of digital speech analysis and synthesis involves a large number of multiplications and therefore requires a high-speed computational capability. Since the speech model is time-varying, the computations are usually carried out in blocks of about 20 or 30 milliseconds' duration. Speech samples are obtained at rates of 8,000 to 10,000 samples per second, with a typical block length of 256 samples.

While one block of samples is being taken, complicated computations such as autocorrelation and updating of digital-filter coefficients must be carried out. A system is typically limited in its capability by the multiplication rates available, since several million multiplications per second often must be performed.

The speech process

The 67558 multiplier permits the building of a generalized processor that can carry out the necessary calculations. By applying rules of speech generation and digital filtering techniques, specialized hardware such as the 67558, under fast microprocessor control, now makes possible relatively low-cost real-time speech processing.

Analysis of the actual speech process is the key to its synthesis. It is generally accepted that speech is created by a combination of glottal excitation—the rapid opening and closing of the glottis, or vocal chords—and noise excitation of the vocal tract produced by the flow of air out of the lungs. The vocal tract is the portion of the windpipe that begins at the glottis and separates into two paths, one of which terminates at the lips and the other at the nostrils. The characteristics of the vocal tract are mainly the cross-sectional area and the resonance coefficients.

In speech processing, it is usually assumed that the source of excitation and the vocal tract system are independent. Accordingly, in speech synthesis the vocal tract is treated as a time-varying digital filter that is excited by pitch impulses to simulate voiced sounds or by random noise to simulate unvoiced sounds. The problem of speech analysis reduces to two parts: computing the numbers that characterize the transfer function of the time-varying digital filter, and determining the position in time of the excitation-pitch pulses (or sometimes only the pitch frequency for periodically occurring excitation pulses).

Speech parameters

Whether impulses or noise should be used as the excitation for the speech-synthesis filter is determined by the voicing. Voiced sounds generate pulses, while unvoiced sounds produce only noise. A model of speech generation is shown in Fig. 6.

Figure 7 is a configuration for digital analysis of speech in order to obtain excitation and filter transfer-function parameters appropriate to a given sample of speech. The parameters are then used to recreate speech in a digital analysis-synthesis communication system. One advantage of digital speech processing is clearly shown by Fig. 8, which illustrates the compression of 64,000 bits per second of encoded speech into a bandwidth of only 2,400 b/s.

A generalized digital processor capable of carrying out calculations for real-time analysis and synthesis of speech either for communication or for research is shown in Fig. 9. □